



RESEARCH AND DEVELOPMENT TECHNICAL REPORT  
CECOM-TR-98-5

**Text Independent Speaker Recognition  
Using A Fuzzy Hypercube Classifier**

**Joseph A. Karakowski and Hai H. Phu**

**October 1998**

Approved for public release;  
distribution is unlimited.

19981020 039

**CECOM  
U.S. ARMY COMMUNICATIONS-ELECTRONICS COMMAND  
RESEARCH, DEVELOPMENT AND ENGINEERING CENTER  
FORT MONMOUTH, NEW JERSEY 07703-5000**

## **NOTICES**

### **Disclaimers**

The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

The citation of trade names and names of manufacturers in this report is not to be construed as official Government endorsement or approval of commercial products or services referenced herein.

# REPORT DOCUMENTATION PAGE

Form Approved  
OMB NO. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comment regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE October 1998	3. REPORT TYPE AND DATES COVERED Technical Report	
4. TITLE AND SUBTITLE TEXT INDEPENDENT SPEAKER RECOGNITION USING A FUZZY HYPERCUBE CLASSIFIER			5. FUNDING NUMBERS	
6. AUTHOR(S) Joseph A. Karakowski and Hai H. Phu				
7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(ES) US Army Communications-Electronics Command (CECOM) Research, Development and Engineering Center (RDEC) Intelligence and Information Warfare Directorate (I2WD) ATTN: AMSEL-RD-IW-TP Fort Monmouth, NJ 07703-5211			8. PERFORMING ORGANIZATION REPORT NUMBER  CECOM-TR-98-5	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)			10. SPONSORING / MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION / AVAILABILITY STATEMENT  Approved for public release; distribution is unlimited.			12 b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words)  The recognition of speakers in an open set, text-independent environment is described. The recognition occurs without any prior training, and occurred in both noisy and clear backgrounds in as little as 1.6 seconds. Investigations and testing were done in the areas of feature characterization of speakers, prefiltering of classifier input, and structure of classifiers for recognition. A prefiltering structure for speech input segments using an expert system implementing hypothesize and test for relevance was investigated. This attempts to maximize classification performance by preselection of most likely voiced speech segments prior to classification. The classifier used was based on Adaptive Resonant Theory and fuzzy Min-Max. It is a neural network with output categories represented by a fuzzy hypercube. The network is described in a hybrid neuronal-functional method. A speaker recognition system was tested using the Switchboard and Greenflag databases. Utterances averaging 0.5 to 7.0 seconds in length were tested, with over 5 hours of conversation for 8, 12 and 16 speaker groups.				
14. SUBJECT TERMS Speaker Recognition; Fuzzy Logic; Signal Processing; Neural Nets; Neural Networks; Speech Recognition			15. NUMBER OF PAGES 43	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT  UL	

# TABLE OF CONTENTS

<b>SUMMARY</b>	<b>vii</b>
<b>1. INTRODUCTION</b>	<b>1</b>
<b>2. FEATURE PROCESSING</b>	<b>3</b>
2.1 Signal Conversion and Formatting	4
2.2 Signal Segmentation	4
2.2.1 Time Segmentation	4
2.2.2 Voiced/Unvoiced Signal Set Partition	6
2.3 Signal Feature Generation	7
2.3.1 LPC Analysis	8
2.3.2 Mel Cepstral Feature	10
2.3.3 Feature Averaging	11
2.4 Feature Selection	11
<b>3. CLASSIFIER PREPROCESSING</b>	<b>12</b>
<b>4. NEURAL NETWORK CLASSIFIER</b>	<b>13</b>
4.1 Basic ART2 Neural Net [13]	13
4.2 Fuzzy ART	15
4.3 Fuzzy Hypercube ART	16
4.3.1 Fuzzy Hypercube ART Structure	17
4.3.2 Fuzzy Hypercube Differences	18
4.3.3 Input Layer	18
4.3.4 Transform Layer	20
4.3.5 Fusion Layer	22
4.3.6 Hypothesize Layer	22
4.3.7 Test Layer	23
4.3.8 Functional Layer	23
4.3.9 Category Layer	25
4.4 Category Merge	26
4.4.1 Merge Parameters	26
4.4.2 Merge Criteria	26
4.5 Initialization	26
<b>5. TEST METHODOLOGY</b>	<b>27</b>
5.1 Test Data	27
5.1.1 Switchboard data set	27
5.1.2 Greenflag data set	28
5.2 Test Conditions	28
5.3 Test Results	29
5.3.1 Feature Processing	29
5.3.2 Neural Network	30
5.3.3 Overall System	30

<b>6. DISCUSSION</b>	<b>32</b>
6.1 Overall	33
6.2 Recommendations for Future Research and Improvements	33
<b>7. BIBLIOGRAPHY</b>	<b>33</b>
7.1 Technical References	33
7.2 Data Base References	35

## LIST OF FIGURES

<b>Figure 1 Time Segmentation of Speech Signal</b>	<b>4</b>
<b>Figure 2 Classifier preprocessing System</b>	<b>12</b>
<b>Figure 3 Basic ART2 Architecture</b>	<b>13</b>
<b>Figure 4 Fuzzy Hypercube ART Layer structure</b>	<b>17</b>
<b>Figure 5 FHNN Functional Diagram</b>	<b>19</b>
<b>Figure 6 Trapezoidal degree of Inclusion</b>	<b>20</b>

## LIST OF TABLES

<b>Table 1 Switchboard 95 Test Set to Actual Speaker Reference</b>	<b>27</b>
<b>Table 2 Greenflag Test Set to Actual Speaker Reference</b>	<b>27</b>
<b>Table 3 Maximum/Minimum Values of Features</b>	<b>28</b>
<b>Table 4 Test Results for 8 Speakers Groups</b>	<b>30</b>
<b>Table 5 Test Results for 12 Speakers Groups</b>	<b>31</b>
<b>Table 6 Fuzzy Hypercube Neural Network Test Results</b>	<b>31</b>
<b>Table 7 Overall System Test Results</b>	<b>32</b>

## LIST OF ACRONYMS

A/D	Analog/Digital
AMDF	Average Magnitude Difference Function
ANN	Artificial Neural Network
ART	Adaptive Resonance Theory
ASR	Automated Speaker Recognition
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DOI	Degree of Inclusion
DPM	Degree of perfect Match
FFT	Fast Fourier Transform
FHNN	Fuzzy Hypercube Neural Network
FV	Feature Vector
HCN	Hypercube Category Nodes
HV	Hypercube Volume
LTM	Long Term Memory
LPC	Linear Prediction Coefficients
MFCC	Mel Frequency Cepstral Coefficients
NN	Neural Network
SRS	Speaker Recognition System
STM	Short Term Memory

## SUMMARY

The recognition of speakers in an open set [19], text-independent environment is described. The recognition occurs without any prior training and in both noisy and clear backgrounds in as little as 1.6 seconds. Investigations and testing were done in the areas of: feature characterization of speakers, pre-filtering of classifier input, and structure of classifiers for recognition.

A feature-based speaker model was used consisting of Linear Prediction Coefficient (LPC) Cepstrum, Reflection Coefficients, and Mel Cepstrum for classification, and energy, pitch, zero crossings for voiced/unvoiced decisions..

A prefiltering structure for speech input segments using an expert system implementing hypothesize and test for relevance was investigated. It attempted to maximize classification performance by pre-selection of most likely voiced speech segments prior to classification.

The classifier used was based on ART [3] and fuzzy Min-Max [25]. It is a neural network with output categories represented by a fuzzy hypercube. A hypothesis and test is performed by the network for overlapping categories where their fuzzy membership representations are interpreted as degrees of typicality, rather than relative [15]. For category control both a vigilance test and overall hypervolume limit test are used. The hypercube limit is extended beyond the unit hypercube(as in [25]) to allow for more "noisy" feature hypercubes. The network has 7 layers: input, transform, process, hypothesize, test, functional, and category. The output is a category layer represented by a fuzzy feature hypercube for each created class. The network is described in a hybrid neuronal-functional method.

A speaker recognition system (based on [12,13]) was tested using the Switchboard [27] and Greenflag [28] data bases. Utterances averaging 0.5 to 7.0 seconds in length were tested, with over 5 hours of conversation for 8 speaker groups, with less time for 12 and 16 speaker groups. The fuzzy hypercube neural network, characterizing one speaker per category, produced an average of 6.29 correct and 0.29 incorrect categories out of a possible 8 total, with no prior training. Overall percent correct classification was found to be 66.9% average for 8 speaker groups.

# 1. INTRODUCTION

The problem of text independent speaker recognition has been of interest to many investigators (see Peacocke [18] for an introduction, Atal [1] for some technical issues.) Markel and Davis [17] obtained text-independent speaker recognition results of 98% correct requiring an average of 39 seconds of speech. The proper choice of signal features for effective speaker recognition is a major issue (see Reynolds [20], Soong [26], Pellisier [19]).

The system described requires recognition to be made with:

- Noisy environment
- Average of 3 seconds of speech
- No prior learning/training

The speakers considered were taken as an "open-set" task [19], where the recognition system had to classify both speakers it had "heard and not heard before." Speaker recognition involving text independent information in an open set environment has had limited success to date using short-time samples [19].

This effort was concerned with recognizing an individual speaker's voice out of a set of voices, in a text-independent and short-time environment. It involved two investigations. First, generation of a descriptive set of voice features sufficient to characterize a speaker in the problem environment, and second, formulation of a reliable classification without any prior training given the feature set based on voiced segments.

A Speaker Recognition System (SRS) [12,13] was used as a test vehicle which accepted either analog or digitized voice signals, and produced a speaker characterization. Feature processing developed a set of descriptive signal features which were classified into speaker classes. This report develops details for the following areas of the SRS.

- Feature Processing
- Classifier Pre-Processing
- Neural Network Classifier
- Test Results

## **Fuzzy ART**

The basic operation of "adaptive resonance" in the standard ART is carried over to the fuzzy ART. The basic equations which govern the fuzzy ART are based on the equations from the standard ART architecture where the intersection operator is replaced by its fuzzy counterpart, the minimum operator. An introduction of the mathematics governing the fuzzy ART is given here, based on Carpenter et al. [2,3,4,5].

The fuzzy ART system consists of three layers: the input layer (F0), processing layer (F1), and output category (F2) layer. Associated between layers F1 and F2 are a set of weights directed from F1 to F2. A fundamental difference between the Fuzzy ART and

prior continuous versions are the simplification of the “resonance criteria” by use of only bottom-up weights in the matching process. The matching process consists of two matching operations:

- Degree which input A matches output category C
- Degree which category C matches input A

For the following, the norm of a vector A, which gives an indication of its “size,” is defined as

$$\|A\| = \sum |a_i| \quad (1)$$

The following operations and data structures are associated with each of these layers:

**Input Layer.** Given an input vector A,  $A = \{a_j\}$  or optionally, with the complement

$$A = \{a_j, a_j^C\}, \quad j = 1, 2, \dots, N_{in} \quad (2)$$

where  $a_j^C = 1 - a_j$  is the complement of  $a_j$ .

The addition of the complement of the input vector has the advantage that A is now self-normalized, using the definition of norm in Eq. 1:

$$\|A\| = \|(a_j, a_j^C)\| = \sum_{j=1}^{N_{in}} a_j + \sum_{j=1}^{N_{in}} (1 - a_j) = \sum_{j=1}^{N_{in}} 1 = N_{in} \quad (3)$$

**Output Layer.** The output layer F2 consists of a set C of  $N_{max}$  active categories,

$$C = \{c_1, \dots, c_{N_{max}}\}$$

Each category vector  $c_j \in C$  has an associated LTM weight set

$$W_j = \{w_{1,j}, w_{2,j}, \dots, w_{N_{in},j}\}$$

**Processing Layer.** A category *Choice Function*  $T_j$  measures the degree which input A is a match to category  $c_j$  and its associated  $W_j$  :

$$T_j = \frac{\|A \cap W_j\|}{\alpha + \|W_j\|} = \frac{\|MIN(A, W_j)\|}{\alpha + \|W_j\|} \quad (4)$$

where  $\alpha > 0$  is a choice parameter.

T is the best category choice, and is calculated as the union of all  $T_j$ .

$$T = \bigcup_j T_j = MAX_j(T_j) \quad (5)$$

There are two possible cases that can occur once a category choice is attempted:

Case 1. Equation 5 produces a choice J. A test is performed on the preliminary choice J to test if it meets a threshold criteria called the vigilance test, where the degree to which the preliminary category matches the input A is compared against a threshold  $\rho$

$$\frac{\|A \cap W_J\|}{\|A\|} = \frac{\|MIN(A, W_J)\|}{\|A\|} \geq \rho \quad (6)$$

If the vigilance criteria of equation 6 is not met, the preliminary choice [J] is said to be "reset," and another category choice according to Eqs. 4 and 5 is made from the set of active categories in C. If the vigilance criteria are met, then the system is said to be in a state of resonance, and the input A is incorporated into category J by the following:

$$w_J^{new} = \beta(A \wedge w_J^{old}) + (1 - \beta)w_J^{old} \quad (7)$$

Fast learning is said to occur when  $\beta = 1$ .

*Case 2. Equation 5 produces no choice.* If no category choice can be made, a new category is created  $C_{N+1}$  with

$$w_{N+1}^{new} = w_{N+1}^{old} = A. \quad (8)$$

Initialization:  $N=0$

A simplified fuzzy ART architecture is described by Kasuba [14].

## 2. FEATURE PROCESSING

The speaker recognition system relies on the underlying model assumptions on which it is based. In this case our model is a heuristic one which loosely follows the Linear Predictive Coefficients (LPC), but includes other features to add fidelity to the spectrum of descriptive power of the system. Prior works in characterizing speaker features have been numerous. Atal [1] identified spectral information and cepstrum parameters for Automated Speaker Recognition (ASR). Columbi [9] provides an overview for both speaker and listener feature models. Other models are the RASTA/PLP [11]. Soong et al. [26] investigated transitional spectral features and stated "instantaneous spectral features carry more speaker relevant information than transitional in ASR." Reynolds [20] investigated several features and widths, and reported "simple cepstral mean removal was the best channel compensation technique for all features" (he tested). Pellisier [19] specifically investigated features in the open set recognition case. He reported that lifted LPC cepstral with normalized log energy appended are optimal for the TIMIT corpus, and LPC reflection with normalized log energy are optimal for the tactical GREENFLAG corpus. "In general, LPC Cepstrum appended or not, perform well." Additionally, [19] found that transitional features did not perform as well as static features, and that decision fusion techniques are the best means of capitalizing on the temporal information. Mel frequency cepstrum also performed well, but not as well as LPC cepstrum and reflection coefficients.

The characterization of a speaker's voice signal into representative features can be broken into several basic phases of processing: Signal Conversion and Formatting, Signal Segmentation, and Feature Processing.

## 2.1 Signal Conversion and Formatting

Voice signal is converted to a digital signal representation for the next stage of segmentation processing. Raw voice signal can be captured by a microphone, from a receiver detector, or other transducer, as well as being provided by a digitized database. This signal is amplified or attenuated, and applied to an A/D converter.

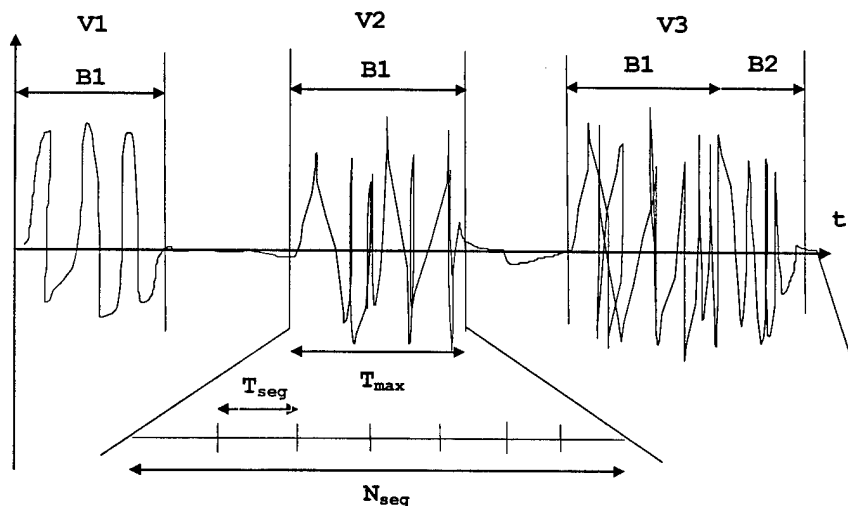
The NIST SPHERE speech format standard was used for control of the voice signal data used in the investigation. In the case of digitized database packages, the NIST/SPHERE voice representation was used as an interface standard. Additionally, it provides conversion information, such as Analog/Digital rates.

## 2.2 Signal Segmentation

Signal Segmentation consists of processing the digital signal to determine suitability for the actual feature space representation and processing. This is accomplished through two separate operations on the signal segments, time segmentation, and voiced/unvoiced signal set partition.

### 2.2.1 Time Segmentation

Time segmentation of the input signal develops a basis for segment to segment processing and averaging over many segments. The segment length is taken from FFT requirements and the sampling rates for the Analog/Digital converter. Speech segments in the range 20-50 ms are created for processing by the system one segment at a time. The segments can overlap by 0-100% of the signal, and tests using different overlaps were performed. A more detailed view of the segmentation process is seen in figure 1. A series of definitions is given in terms of signal processing.



Time Segmentation of Speech Signal  
Figure 1

**Signal.** Time function resulting from signal conversion and formatting operation of section 2.1. The unprocessed signal  $V$  has the basic characteristics of being non-periodic, bounded, energy limited, duration limited, and band limited:

$$V(T) = \{x; x(t+T) \neq x(t), \quad -\infty < t < \infty\} \quad [\text{non-periodic}]$$

$$V(K) = \{x; |x(t)| < K, \quad -\infty < t < \infty\} \quad [\text{bounded}]$$

$$V(K) = \{x; \int_{-\infty}^{\infty} x^2(t) dt < K\} \quad [\text{energy-limited}]$$

$$V(T) = \{x; x(t) = 0 \text{ for all } |t| > T\} \quad [\text{duration-limited}]$$

$$V(W) = \{x; X(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi ft} dt = 0 \text{ for all } |f| > W\} \quad [\text{band-limited}]$$

where  $K$  is a positive, real number,  $T$  is a period,  $W$  is frequency band

**Burst Signal.** A burst  $B_i^j$  of a signal  $V_j$  is a consecutive individual duration-limited segment of a signal, in a series of one or more non-overlapping segments. The signal set  $V$  contains all the signals over a time period of interest.

$$V = \bigcup_j V_j, \quad V_j = \bigcup_i B_i^j \quad \text{where } j=1,2,\dots,N_v \text{ and } i=1,2,\dots,N_b \quad (9)$$

$$B_{i1} \cap B_{i2} = 0 \quad i1 \neq i2 \quad \text{over all } i$$

$N_v$  is the number of signals,  $N_b$  is the number of bursts in signal  $j$ . All the  $V_j$  are assumed to be independent. Each burst is composed of a series of non-overlapping segments. The following criteria on the bursts hold:

- The segments  $S_k^i$  of the burst  $B_i^j$  are predominantly from the set of voiced segments.
- The burst length is limited to a maximum value  $T_{\max}^B$ .

A relation bounding the number of segments  $N_{seg}^i$  in any burst  $[i]$  is defined as

$$T_{\max}^B \leq N_{seg}^i T_{seg} \quad \text{for all } i=1,2,\dots,N_b \quad (10)$$

where  $T_{seg}$  is the segment constant time, and  $N_{seg}^i$  is the number of segments in burst  $i$

**Segment Overlap.** Each of the  $N_{seg}^i$  segments in burst  $B_i^j$  is said to uniformly overlap if each consecutive segment has  $[jj]$  samples of signal in common with the previous segment. If the number of samples in a segment is  $N_{sam}$ , we have the following relation for the degree of overlap,  $D_{ol}$ :

$$D_{ol} = \frac{jj}{N_{sam}} \quad (11)$$

The SRS testing varied  $D_{ol}$  from 0-50%.

### 2.2.2 Voiced/Unvoiced Signal Set Partition

A voiced/unvoiced partition of the signal segment set is made through an algorithm based on [1]. This set partition is made using elementary signal features such as average zero crossings [22], average pitch [21], and average log energy [22]. The methods to develop each are described below.

**Pitch** The Average Magnitude Difference Function (AMDF) [21] is used for pitch extraction. It is a variation on autocorrelation analysis where, instead of correlating the input speech at various delays, a difference signal is formed between the delayed speech and the original. At each delay, the absolute magnitude of the difference is taken. At delay = 0, the difference signal is always zero but exhibits deep valleys at delays corresponding to the pitch period of voiced sounds. The AMDF pitch extractor was chosen because it gives good estimation of pitch contour and requires no multiply operations as in the autocorrelation method, thus improving efficiency. The following is the AMDF algorithm for extracting the Pitch Period per segment of speech:

Step 1. Using the Difference relation in (1), find the AMDF for delay  $n \geq 0$ , where  $n = 0, 1, \dots, N_{sam}$ ,

$N'$  = number of samples in the subset of the chunk,  
 $S_k$  is a sample from the original signal,  
 $S_{k-n}$  represents a sample from signal delayed by  $n$ .

$$N' = N_{sam} * 0.75$$

$$D_n = \frac{1}{N'} \sum_{k=n}^{N'-1} |S_k - S_{k-n}| \quad (12)$$

A percentage of samples in Eq. 12 of 75% were used in the AMDF correlation.

Step 2. From the AMDF find the first pitch valley where  $n > 0$ . The delay at the point of the valley is the pitch period. The inverse of the pitch period is the pitch  $P$  of the voiced speech in frequency.

**Average Zero Crossings.** The average zero crossings is determined from the number of sign changes in a signal segment over time. A count  $C$  is made over the entire segment length  $T$  by counting the number of times the following occurs between each sample  $x(n)$  and sample  $x(n-1)$  in the segment,

$$\text{sign}[x(n)] \neq \text{sign}[x(n-1)] \quad (13)$$

The average zero crossing  $n_z$  is equal to

$$n_z = \frac{2f_s}{T} C \quad (14)$$

Since the energy of voiced speech signal is concentrated below 3 kHz, and the energy of fricatives is generally above 3 kHz, zero crossing information can be used as a feature in voice/unvoiced speech characterization [21].

**Average Log Energy** is another signal measure used for voiced/unvoiced detection. It is computed on each speech segment. The energy calculation is given by [22]

$$E_{\log} = 10 \log_{10} \left( \frac{1}{N} \sum_{i=1}^N x_i^2 \right) \quad (15)$$

**Voiced/Unvoiced Rule.** The Voiced/Unvoiced characterization of a single segment is a majority-based decision using criteria of the pitch, zero crossings, and Average Log Energy of each input segment. The following algorithm was used:

**Voiced/Unvoiced Algorithm:** Given:  $P, n_z, E_{\log}$  for a segment

```

Step 1:  $M=0$ 
Step 2: IF  $n_{\min} \leq n_z \leq n_{\max}$  THEN  $M=M+1$ 
Step 3: IF  $E_{\log} \geq E_{\min}$  THEN  $M=M+1$ 
Step 4: CASE M
        2: "Voiced"
        0: "Unvoiced"
        1: IF  $P_{\min} \leq P \leq P_{\max}$  "Voiced"
           ELSE "Unvoiced"

```

where

$P_{\min}, P_{\max}$  are the minimum and maximum pitch, 30-500 Hz

$n_{\min}, n_{\max}$  are the minimum (30) and maximum (3000) zero crossing frequency

$E_{\min}$  is the minimum voiced energy threshold

If the majority of the tests are true then the speech segment is assumed to be voiced. Otherwise the segment is assumed unvoiced and is discarded.

### 2.3 Signal Feature Generation

The feature processing calculates various signal transform features which represent different characterizations of a speaker through his voice signal. Linear Prediction Coding finds the coefficients from the Inverse Filter,  $A(z)$ , defined by Markel [16]. The significance of the Inverse Filter is that it can realize a model of the physical speech production system such as the Glottal  $G(z)$ , the Vocal Tract  $V(z)$  and the Lip Radiation  $L(z)$  system [3].

$$\begin{aligned} A(z) &= 1 + \sum_{i=1}^p a_i z^{-i} \\ &= 1/G(z)V(z)L(z) \end{aligned} \quad (16)$$

The signal feature processing is performed in three consecutive phases: a) LPC Analysis, b) Mel Cepstrum Calculation and c) Feature Scaling.

### 2.3.1 LPC Analysis

The LPC analysis consisted of setting filter constants and initialization parameters, followed by Pre Emphasis, Hamming Window, Auto Correlation, D'Urbin expansion (LPC/autocorrelation and Reflection Coefficients), LPC Cepstrum, and Delta Cepstrum.

#### 2.3.1.1 Pre-Emphasis:

A given segment of speech is pre-emphasized by the following function.

$$w(i) = s(i) - 0.98s(i-1), \quad i = 1, 2, \dots, N_{sam}, \quad (17)$$

where  $s(i)$  is a sample in a segment

#### 2.3.1.2 Hamming Window:

The use of a Hamming window reduces effects of oscillations and poor convergence.

$$w(i) = \begin{cases} |i| \leq \frac{N}{2}, & 0.54 - [0.46 \cos \frac{2\pi i}{N}] \\ \text{otherwise,} & 0 \end{cases} \quad (18)$$

#### 2.3.1.3 Autocorrelation Coefficients:

The auto correlation coefficients  $C(i)$  are determined by:

$$C(i) = \frac{1}{N} \sum_{j=0}^{N_{sam}-i} S(j)S(j+i) \text{ for } i=1, \dots, O_{corr} \quad (19a)$$

normalizing,

$$C(i) = \frac{C(i)}{C(0)} \quad \text{for } i=1, \dots, O_{corr} \quad (19b)$$

where  $O_{corr}$  is the correlation order.

#### 2.3.1.4 D'Urbin Expansion:

Function to compute LPC parameters with D'Urbin's formula. The LPC and reflection coefficients are calculated using the autocorrelation coefficients  $C(i)$ .

##### D'Urbin's formula:

1. Initialization

$$lpc_1 = 1.0$$

$$lpc_2 = -\frac{c_1}{c_0}$$

$$\alpha = c_0[1 - lpc_2^2]$$

2. Algorithm

DO FOR  $i=2$  TO  $O_{lpc}$

$$r_i = \frac{-\sum_{j=1}^{j \leq i} lpc_j * c_{i-j+1}}{\prod_{k=2}^i (1-r_i^2)(1-r_i^2)}$$

FOR j=2 TO O<sub>lpc</sub>

$$A_j = lpc_j + r_i * lpc_{i-j+2}$$

$$lpc_j = A_j$$

$lpc_0 = 1.0$

FOR j=1 TO O<sub>lpc</sub>

$$lpc_j = -lpc_{i+1}$$

### 2.3.1.5 LPC Cepstrum

The Cepstrum [9] is, by definition, the inverse Fourier transform of the logarithm of the transfer function. The Cepstral Coefficients were obtained directly from the LPC coefficients. Atal defines the cepstrum as the inverse Fourier transform of the logarithm of the transfer function [1].

$$\ln H(z) = C(z) = \sum_{k=1}^{\infty} c_k z^{-k}$$

The all pole filter model based on predictive analysis on speech samples is

$$H(z) = \frac{G}{1 + \sum_{k=1}^p lpc_k z^{-k}}$$

It can be shown that, given the all-pole model, a recursive relation exists between the cepstral coefficients  $c_k$  and the predictor coefficients  $a_k$ .

$$c_1 = lpc_1$$

$$c_k = \sum_{l=1}^{k-1} \left(1 - \frac{l}{k}\right) lpc_l c_{k-l} + lpc_k, \quad 1 < k \leq p \quad (20)$$

$$p = \frac{f_s}{1000} + \gamma \quad \gamma \approx 3$$

The sampling frequency  $f_s$  determines the number of poles, modified by a fudge factor.

### 2.3.1.6 Delta Cepstrum (from [26])

Given  $c_m$  and  $c'_m$ , the cepstral representations of two bursts, the delta cepstrum is found for the first  $p$  cepstral coefficients:

$$d_{CEP} = \sum_{m=1}^p (c_m - c'_m)^2 \quad (21a)$$

In order to equalize the contributions from individual cepstral components, a weighted cepstral distance is desirable. Using the Mahalanobis distance, and since the estimated covariance matrix is essentially diagonal, we obtain:

$$d_{WCEP} = \sum_{m=1}^p (c_m - \bar{c}_m)^2 w_m \quad (21b)$$

where the weighting coefficient  $w_m$  is the reciprocal of the variance of the  $m$ th cepstral coefficient.

The generalized slope in time has the following form:

$$\Delta c_m(t) = \frac{\sum_{k=-K}^K k * h_k * c_m(t+k)}{\sum_{k=-K}^K h_k * k^2} \quad (21c)$$

### 2.3.2 Mel Cepstral Feature

Linear prediction cepstral coefficients generated from the LP spectrum and distributed along a linear frequency axis, form a less than optimal representation of an auditory signal since a logarithmic function of frequency better approximates the ability of the human ear to discriminate frequencies. The Mel scale is often used to approximate the resolution of the human auditory system's perception of speech. Deller et al. defines the Mel as "a unit measure of perceived pitch or frequency of a tone." An equation for approximating the Mel scale is:

$$F_{mel} = \frac{1000}{\log(2)} \log(1 + F_{Hz}/1000)$$

The Mel frequency cepstral coefficients (MFCC) are obtained by Mel warping the spectrum's frequency scale before taking the fast Fourier transform

$$\text{Mel cepstrum} = \text{FFT}(\log|\text{Mel spectrum}|)$$

**Development of Mel Cepstrum.** The Mel cepstral coefficients are generated by the following procedure:

1. Calculate the Mel Bands: The Mel bands are calculated from  $N_{bands}$  the number of Mel bands, and the start and end frequencies,  $f_{start}$  and  $f_{end}$ ,

$$f_{start,end}^{mel} = 2595 \log\left(1 + \frac{f_{start,end}}{700}\right)$$

$$step = \frac{f_{end}^{mel} - f_{start}^{mel}}{N_{bands}}$$

Each band is a multiple  $n$  of the step in Mel frequency and is calculated by:

$$band_n^f = \left(10^{\frac{f_{start}^{mel} + n * step}{2595}} - 1\right) * 700$$

Each band is converted to the integer value of the sample to which it corresponds,

$$band_n^{int} = \left[ band_n^f - band_0^f \right] \left( \frac{N_{sam}}{(band_{N_{bands}}^f - band_0^f)} \right) + 0.5$$

2. Weight Bands: A square filter is used to weight the bands. It is an all pass function for each of the Mel bands.
3. Preemphasize: See section 2.3.1.1 above
4. Hamming Window: See section 2.3.1.2 above.
5. Fast Fourier Transform (FFT): The FFT for the discrete signal with  $N_{samp}$  points, which is a power of 2, which produces the discrete fourier transform  $dft_n$
6. Weighted cepstral: The magnitude of the DFT is weighted by the appropriate weight for the band and the inverse log taken to form the cepstrum.

$$c_l = \log \left( \frac{w_m |dft_m|}{band_{l+1}^{int} - band_l^{int}} \right) \quad (22)$$

7. Discrete Cosine transform (DCT): The DCT is performed on the weighted cepstral components to obtain the final result.

$$ct_l = \sum_{m=0}^{N_{filter}} l * c_m \cos \left( \frac{l\pi(m-0.5)}{N_{filter}} \right) \quad l=0,1,2,\dots,M \quad (23)$$

### 2.3.3 Feature Averaging

An averaging of each of the features was done. Each individual feature is averaged over all the features for each of the  $N_{seg}$  segments,

$$f_{unscaled}^i = \frac{\sum_{j=0}^{N_i^{max}} f_j^i}{N_j^{Max}} \quad (24a)$$

## 2.4 Feature Selection

The feature sets selected for final implementation for speaker recognition were based on the results of Pellissier [19]. The set utilized was

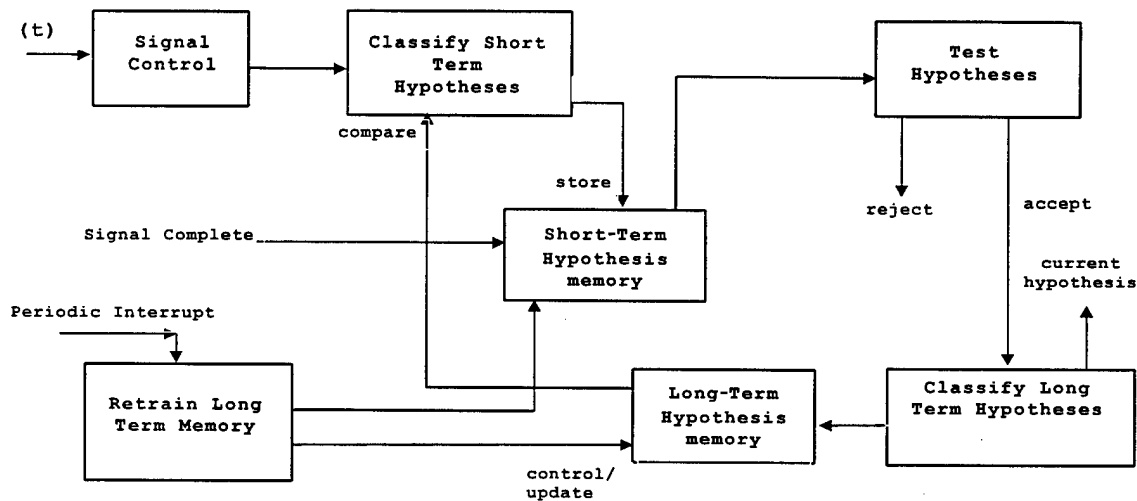
- LPC Cepstral (7 coefficients), defined by equation (20).
- Reflection coefficients (12 coefficients), defined by D'Urbin expansion in 2.3.1.4
- Mel Cepstral (13 coefficients), defined by equation (22)

Additional features considered during testing:

- Delta Cepstrum
- Pitch
- Energy
- Listener Model

### 3. CLASSIFIER PREPROCESSING

A series of experiments was performed to assess the usefulness of preprocessing speech feature data for the classifier. The overall structure for preprocessing is a test structure which develops a set of short-term hypotheses about the current signal and tests to determine which segments of the signal should be passed on to the actual classifier or to long-term hypothesis memory. A block diagram of the preprocessing scheme is shown in figure 2.



**Classifier Preprocessing System**  
**Figure 2**

The hypothesis and test paradigm was investigated to select information to be learned by a Neural Network Classifier and reject information that was unsuitable. The criteria of the selection are made on the basis of the intersegment global information structure. The segment data are rated according to their:

- 1) overall rating similarity
- 2) grouping of like versus unlike segments in time.

The overall rating similarity was done by class average results of the preclassification process, i.e., for each potential class, an average of the result was given,

$$avg = a(i) / \sum a$$

For the grouping of like terms, a network of all segments in a "group" of segments, which is a related unit, are compared in their time relationship to each other. Thus, if two segments next to each other are of like pre-class, the linkage is strong, whereas, if two segments are separated by an unlike segment, they have less linkage and so on. A directed graph of relations was created and used to rate linkage strength.

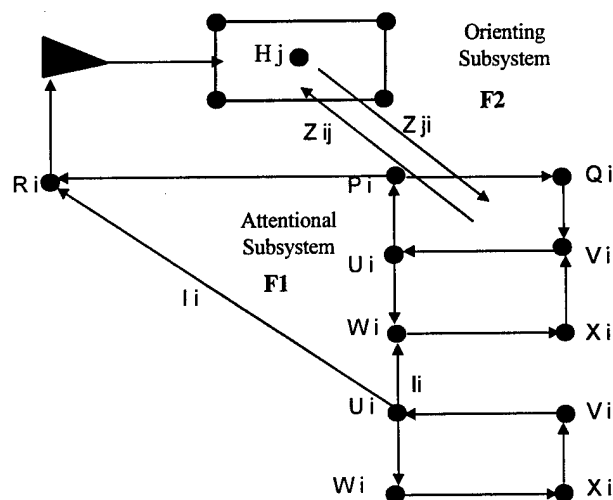
The incoming signal was tested by subjecting it to a series of classifications that are stored in short-term memory (STM). After the series are completed, the contents of the STM are tested according to the grouping and linkage criteria using an expert system. The results of the test determine if the STM contents are allowed to retrain the long-term memory (LTM). Each data point is retrained and if accepted by the test, reclassified, and finally stored. Additionally, an optional periodic retraining of long term memory using all accepted signals over a finite period is done to eliminate any long term averaging effects on the individual speaker signatures. An optional output of each result of the current hypothesis is available for further processing.

## 4. NEURAL NETWORK CLASSIFIER

The recognition of a speaker from a set of features requires a clustering/classification process which is able to form any number of classes dynamically, and tolerate the noisy and overlapping domain of speaker feature vectors. In this effort, ART model [3-7] was used to cluster and classify unknown speakers. There were three networks considered during this investigation: ART [3,10,12], fuzzy ART [7,11], and fuzzy hypercube ART [25]. After some preliminary testing of all three networks, emphasis was placed on modification of fuzzy ART neural network architecture for speaker recognition.

### 4.1 Basic ART2 Neural Net [13]

The general operation of the basic ART2 neural network architecture is described. This forms the basis for the fuzzy ART and fuzzy hypercube ART networks. A typical ART2 neural network is composed of two layers of fully interconnected neurons. Adaptive connections between neurons store long term memory (LTM) traces in the network. LTM represents information that the network has learned. Figure 2 shows basic architecture for ART2 neural network.



Basic ART2 Architecture  
Figure 3

The two layers (or fields) of neurons in an ART2 architecture in figure 3 form the *Attentional Subsystem*. The first field is named the *Feature Representation Field*, or F1. Each F1 neuron contains processing elements that form three intra-PE sublayers which are responsible for processing one element in the input pattern.

The main function of the feature representation field is to enhance the current input pattern's salient features while suppressing noise [23]. This is achieved through pattern normalization and thresholding which are required for the processing of analog patterns. Normalization compares the input pattern and the patterns stored in the network's LTM traces. Thresholding maps the infinite domain of the input patterns to a prescribed range [23]. The second layer in the attentional subsystem is called the *Category Representation Field*, or F2. Each neuron in this field represents one category (or class) that has been learned by the network. The connections from a particular F2 neuron store the *pattern* of the category it represents.

ART2 utilizes an unsupervised competitive learning technique in which patterns are represented by points in an N-dimensional feature space. Pattern similarity is assessed on the basis of a Euclidean distance which states that: *Patterns that are sufficiently close to one another are placed in the same category.*

The N-dimensional centroid location represents that class' exemplar. An unsupervised learning procedure attempts to discover the distributions and centroids of the categories for the patterns it is presented.

ART2 utilizes a "winner-take-all" classification strategy, such as MAXNET, that operates in the following manner:

- (1) An input pattern is presented to the feature representation field where it is normalized and thresholded,
- (2) The resultant signal, which is called short term memory (STM), is passed through bottom-up connections to a category representation field,
- (3) Each established class in F2 responds to the signal with an activation level which it sends to itself through excitatory connections and to all its neighbors through inhibitory connections,
- (4) Eventually the F2 neuron with the highest activation will inhibit the others. The sole remaining active F2 neuron is assumed to most resemble the current input pattern.

Having selected the winner, the *Orienting Subsystem* is activated and determines whether the winning neuron's LTM traces sufficiently resemble the STM pattern to be considered a match. The degree of match between the two patterns is related to the cosine of the angle between them in feature space. Patterns that are very similar are nearly parallel to each other while dissimilar patterns are orthogonal to each other. A matching threshold called the *Vigilance Parameter* determines how similar the input pattern must be to the exemplar to be considered a match [24]. If the degree of match computed by the orienting subsystem exceeds the vigilance parameter, a state of resonance is attained and the STM pattern at F1 is merged onto the winning neuron's LTM traces. Otherwise, the

orienting subsystem sends a reset signal to the winning neuron, and inhibits it from competing again for the current input pattern [23]. This search process is repeated until either an F2 neuron passes the vigilance test or all established F2 neurons have failed the test. In the latter case, a new category is established in the next available F2 neuron.

Learning is considered to be competitive since each F2 neuron attempts to include the current input pattern in its category code. The actual learning process, whereby the current input pattern is encoded into the network's memory, involves modification of the bottom-up and top-down LTM traces that join the winning F2 neuron to the feature representation field. Learning either refines the code of a previously established class, based on any new information that is contained in the input pattern, or initiates code learning in a previously uncommitted F2 neuron [3]. In either case, learning only occurs when the system is in a resonant state. This property ensures that an input pattern does not obliterate information that has been previously stored in an established class. A basic ART architecture was used in prior recognition efforts with some success [12].

## 4.2 Fuzzy ART

The basic operation of "adaptive resonance" in the standard ART is carried over to the fuzzy ART. The basic equations which govern the fuzzy ART are based on the equations from the standard ART architecture where the intersection operator is replaced by its fuzzy counterpart, the minimum operator. Several of the operations are different, however. The top-down and bottom-up matching processes are combined, since the matching between input and category is the same in both directions.

An introduction of the mathematics governing the fuzzy ART is given here based primarily on Carpenter & Grossberg [5, 6, 7]. This will utilize the fuzzy hypercube ART, along with modifications and additions in the next section.

The fuzzy ART system consists of three layers: the input layer (F0), processing layer (F1), and output category (F2) layer. Associated between layers F1 and F2 are a set of bi-directional weights denoted bottom-up, directed from F1 to F2, and top-down, directed from F2 to F1. The following operations and data structures are associated with each of these layers:

$$\begin{aligned} \text{F0: } A_i &= a_i \quad i = 1, 2, \dots, M \quad \text{and, optionally,} \\ A_i &= a_{i-M}^c \equiv (1 - a_{i-M}) \quad i = M + 1, \dots, 2M \end{aligned} \quad (25a)$$

where M is the number of input components with optional complementation and number of category nodes N.

Note that, if the complement is added to A, that the complement coded inputs are self-normalized:

$$|A| = |(a, a^c)| = \sum_{i=1}^M a_i + \sum_{i=1}^M (1 - a_i) = M \quad (25b)$$

$$\begin{aligned} \text{F1: } \bar{x} &= (x_1, \dots, x_{2M}) \\ \bar{x} &= \begin{cases} A, & F_2 \text{ inactive} \\ A \wedge w_j, & J^{\text{th}} F^2 \text{ node chosen} \end{cases} \end{aligned} \quad (26a)$$

through choice function (27b)

The choice is made as “final” if the preliminary choice  $x$  meets a threshold criterion called the vigilance test,

$$\begin{aligned} \frac{|\bar{x}|}{|A|} &\geq \rho, \text{ or} \\ \frac{|A \wedge w_j|}{|A|} &\geq \rho \end{aligned} \quad (26b)$$

If the vigilance criterion is not met, the preliminary choice  $[J]$  is said to be “reset,” and another choice is made from the set of active categories in  $y$ . If there are no more active categories, a new category is created.

**F2:**

$$\begin{aligned} \bar{y} &= (y_1, \dots, y_N) \text{ with associated} \\ \bar{w}_j &= (w_{j1}, \dots, w_{j,2M}) \text{ weights(LTM)} \end{aligned} \quad (27a)$$

The category Choice Function  $T_j$  is defined as:

$$T_j = \frac{|A \wedge w_j|}{\alpha + |w_j|}, \quad (27b)$$

where  $\alpha > 0$  is a choice parameter, and the norm is defined as

$$|p| = \sum_i |p_i| \quad (27c)$$

The category choice is made on the basis of a maximum function,

$$T_j = \max\{T_j\} \quad (27d)$$

If the choice of category made in (17d) passes the vigilance test of equation (16b), then the category is accepted and learning of the weights occurs as follows:

$$w_j^{\text{new}} = \beta(A \wedge w_j^{\text{old}}) + (1 - \beta)w_j^{\text{old}} \quad (27e)$$

Fast learning is said to occur when  $\beta = 1$ .

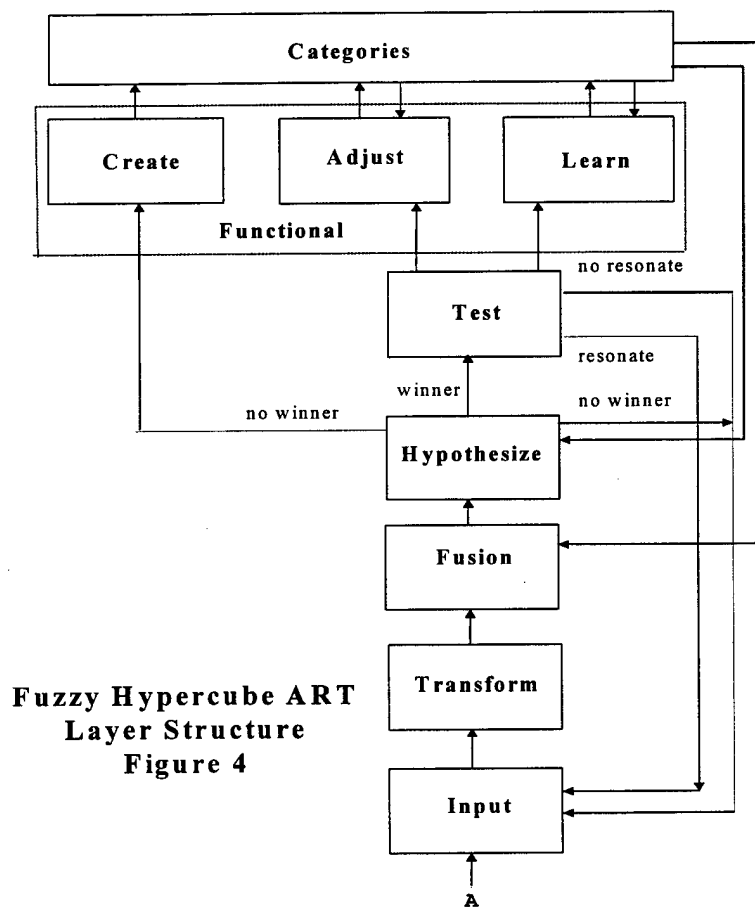
### 4.3 Fuzzy Hypercube ART

The ART neural architectures described in sections 4.1 and 4.2 both did not perform well during speaker recognition testing. They generally suffered from poor tolerance to noise. Modifications of the Fuzzy ART were done to improve performance. Several basic ideas were implemented. One was the current representation of the output categories as hypercubes. An overall volume parameter bounded each hypercube volume. In order to provide some noise tolerance, the hypercubes were additionally fuzzified. Several other

basic functions were extended, including the category choice function, the inclusion of hypervolume limits, and the generalization of the learning algorithm with fuzzy hypercubes. A general overview will be given of the network layer structure, with a more detailed functional description.

#### 4.3.1 Fuzzy Hypercube ART Structure

The fuzzy hypercube neural network has seven layers of processing. Figure 4 shows their interconnection. Each of the layers is briefly described below. One specific item to notice is that the network is both feedforward and feedback. Specific category information is fed back to the Hypothesize and Fusion layers for hypothesis formation, as well as to the Functional layer in category adjustment and learning. Additionally, the resonate/no resonate is an enable/inhibit signal which effectively cycles the entire network in processing data sets synchronously.



**Fuzzy Hypercube ART  
Layer Structure  
Figure 4**

Input: fuzzified and optional functional expansion, equations (15a,b).

Transform: Category choice functions are evaluated over active categories.

Fusion: The category choice functions are fused to final ratings.

Hypothesize: A final category rating is chosen as a “hypothesis,” otherwise, a new category is created.

Test: A vigilance pass/fail test is performed matching input to chosen category.

Functional: Categories are created, hypervolume adjusted, or input learned.

Category: Hypercube feature vectors, and control .

#### 4.3.2 Fuzzy Hypercube Differences

The fuzzy hypercube ART neural network has several distinct differences from the basic ART and fuzzy ART. It retains the basic data structures using A and X vectors. The concepts of bottom-up and top-down match as well as the learning rules are very different. The fuzzy hypercube layers and the differences between prior ART architectures and the current one will be described in the following sections. A detailed view of the network is shown in figure 5, where each of the blocks from figure 4 are broken down to the next level of details.

#### 4.3.3 Input Layer

The Input layer has several inputs and outputs. An enable/disable set of inputs effectively controls the resonance of the network. The network is either allowed to continue cycling through with the current input A, when a suitable category is not found by the Hypothesize/Test layers, or to stop the current input and enable the acceptance of the next input upon finding a suitable category (or creating a new one).

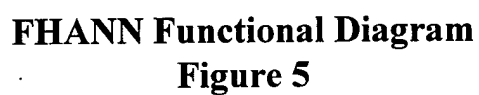
The Input layer, if enabled by a “no resonate” signal, fuzzifies and optionally expands the input information. Each input dimension in A is translated into a fuzzy membership function onto the [0,1] interval, which indicates the degree of absence, by its nearness to its lower bound, or presence, by nearness to its upper bound. The translation is a mapping  $F = \{f\} \rightarrow [0,1]$

This operation requires a pre-learning of the maximum  $F_{Max}^i$  and minimum  $F_{Min}^i$  expected value for each individual feature. For each feature i,  $f_{unscaled}^i$ , we scale it to

$$f_{scaled}^i, \text{ by: } f_{scaled}^i = \frac{f_{unscaled}^i}{|F_{Max}^i - F_{Min}^i|} \quad (28)$$

The set of scaling coefficients,  $|F_{Max}^j - F_{Min}^j|$ , for each feature [j] can be considered as weighting factors, determined by some learning function, but in the form of the difference between two quantities, not absolute values.

The method of determining the values of  $F_{Max}^i$  and  $F_{Min}^i$  were not performed during normal operation of the neural network, but off line, and provided as inputs to the process. The values  $F_{Max}^i$  and  $F_{Min}^i$  were experimentally determined from observation of the maximum and minimum values of each of the features [j]. Note that outlier feature values outside of the given scaling ranges are normalized to 0.0 or 1.0 to indicate either full membership, or no membership in the feature set.



#### 4.3.4 Transform Layer

Membership values from the Input layer are passed to the Transform layer, where they generate two membership functions for each active category node, a “Degree of Inclusion” (DOI), and a “Degree of Perfect Match” (DPM). These memberships together give an indication of the degree to which the input matches each feature category hypercube. The development of memberships is done through a fuzzy procedure (see Eqs. 31-34).

The choice function (Eq. 27b) has been expanded by Carpenter and Gjaja [8] to Choice-by-difference. Simpson [25] develops a membership function which measures the degree to which an input  $A$  fits within the hypercube defined by Eq. 33. He defines a function  $b_j$ , which approaches 1 as the point gets nearer to the hypercube,

$$b_b(A_h, V_j, W_j) = \frac{1}{n} \sum_{i=1}^n [1 - f(a_{hi} - w_{ji,\gamma}) - f(v_{ji} - a_{hi,\gamma})] \quad (29)$$

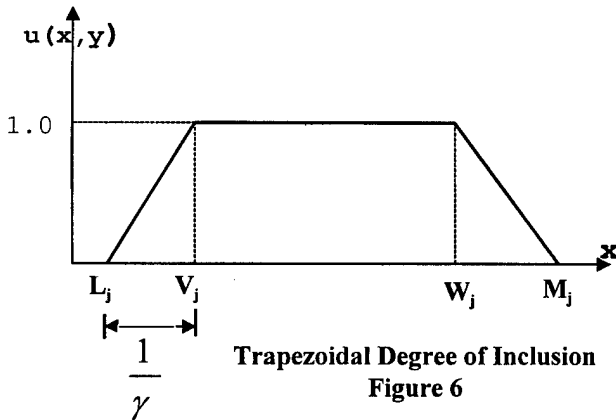
where  $f()$  is the ramp function,

$$f(x, y) = \begin{cases} 1 & \text{if } x\gamma > 1 \\ x\gamma & \text{if } 0 \leq x\gamma \leq 1 \\ 0 & \text{if } x\gamma < 0 \end{cases} \quad (30)$$

The choice function is generalized to a hypercube match. The choice function is defined by two related nonlinear functions, the degree of inclusion and the degree of perfect match, which are developed in parallel, and combined by a fusion function.

#### Degree of Inclusion.

The degree of inclusion (DOI) function measures the level to which each dimension of the input  $A_j$  is inside a category hypercube.



DOI is a trapezoidal membership function which gives full membership whenever an element of  $A_j$  is included in a category, and less than full membership outside, depending on the distance to the hypercube. Figure 6 describes the shape of the membership function.

The membership for DOI,  $\mu^{DOI}(x)$  is defined for each dimension of a hypercube  $H_j$ :

$$H_j = \{h_j^1, h_j^2, h_j^3, h_j^4\} \quad (31)$$

$$\mu_j^{DOI}(x) = \begin{cases} 1 & \text{if } x \geq h_j^2 \text{ or } x \leq h_j^3 \\ \frac{1.0 - (x - h_j^3)}{|h_j^3 - h_j^4|} & \text{if } h_j^4 \geq x > h_j^3 \\ \frac{1.0 + (x - h_j^2)}{|h_j^1 - h_j^2|} & \text{if } h_j^1 \leq x < h_j^2 \\ 0 & \text{if } x > h_j^4 \text{ or } x < h_j^1 \end{cases} \quad (32)$$

Usually,  $|h_j^1 - h_j^2| = |h_j^3 - h_j^4|$  to evenly fuzzify the hypercube. The overall membership function  $\mu^{DOI}(x)$  is the sum of the individual memberships:

$$\mu^{DOI}(x) = \sum_j \mu_j^{DOI} \quad (33)$$

**Degree of Perfect Match.** The measure of the distance from the mean of each dimension of  $H_j$  is defined as the degree of perfect match (DPM). The DPM is a similarity relation between the input  $x$  and an individual category. The dissimilarity is defined as the difference between the value  $x$  and the mean of the category  $x$ ,  $m_j$ :

$$Dissimilarity_j \cong |x - m_j| \quad (34a)$$

The similarity is the complement of the dissimilarity,

$$Sim_j = Dissimilarity_j^c = 1 - |x - m_j| \quad (34b)$$

The membership for DPM,  $\mu^{DPM}(x)$  is defined for each dimension of a hypercube  $H_j$  and is derived as follows. The mean of each dimension is

$$m_j = \frac{|V_j - W_j|}{2} \quad (34c)$$

and the membership function for each dimension  $j$  is

$$\mu_j^{DPM}(x) = \begin{cases} 1 - |x - V_j| * P(x) & \text{if } W_j = V_j \\ 1 - |x - m_j| * P(x) & \text{if } W_j \neq V_j \end{cases} \quad (34d)$$

where  $P(x)$ , the possibility of vigilance is defined as,

$$P(x) = \begin{cases} 0 & \text{if } x \geq T_d \\ & \text{if } 1 > \frac{x - T_d}{T_d} > 1 \\ & \text{else} \\ d & \end{cases} \quad (34e)$$

The overall membership function  $\mu^{DPM}(x)$  is the sum of the individual values,

$$\mu^{DPM}(x) = \sum_j \mu_j^{DPM} \quad (34f)$$

#### 4.3.5 Fusion Layer

The DOI and DPM from the Transform layer, as well as certain feedback category information, comprises the input to the Fusion layer. The fusion of the membership functions for degree of inclusion and degree of perfect match are done with a dynamic weighting and normalization of the two functions. The dynamic weighting is done to compensate for low DOI at the start of a matching process

$$R^J(x) = k_1 \mu^{DPM}(x) + k_2 \mu^{DI}(x) \quad (35a)$$

where

$$\begin{aligned} k_2 &= \min(k_2 * NC, 1) \\ k_1 &= k_2^C \end{aligned} \quad (35b)$$

and NC, the node constant, is a dynamic weighting function defined as:

$$NC(j) = \begin{cases} 0.65, & N_c = 1 \\ 0.85, & N_c = 2 \\ 0.95, & N_c = 3 \\ 1.00, & N_c > 3 \end{cases} \quad (35c)$$

#### 4.3.6 Hypothesize Layer

The inputs from the Fusion layer form a number of potential hypotheses from which a single hypothesis is chosen. The hypothesis is formed by a maximum over all the input possibilities.

$$\begin{aligned} \text{Winner; } C_k &= \max_k \{R^k\} \text{ if } R^k \text{ is active and } R^k > 0 \\ \text{NoWinner, if } R^k &\text{ is inactive over } 0 < k \leq n \end{aligned} \quad (35d)$$

The resultant hypothesis of Winner is passed with the winning category node to the Test layer, while the No Winner Hypothesis is passed back to the Input layer to halt resonation of the network, as well as to create a new category node for the current input A.

#### 4.3.7 Test Layer

The Test layer performs the vigilance test on the current input  $A$  and the category input hypothesis. The vigilance test, in the standard and fuzzy ART, is a vector matching process as shown in Eq. 29b. In the fuzzy hypercube ART, the vigilance test is a general test for category hypercube membership. The test is performed using a modified form of Eq. 16b,

$$\frac{|\bar{x}|}{|A|} \geq \rho^{adj}, \quad \text{or} \quad \frac{|A \wedge w_J|}{|A|} \geq \rho^{adj}$$

where  $\rho^{adj} = \rho g(n)$ ,  $g(n)$  is the vigilance adjustment function, and  $n$  is number of times a category is visited.

The Test layer has several outputs depending upon the result of the vigilance test. If a category passes the test, the Input layer is signaled to halt resonation of the network, and that category is passed to the Functional Layer. Additionally, the category layer is re-enabled for all nodes to compete in hypothesizing and testing of the Fusion and Hypothesis layers.

In the case when a category fails the vigilance test, the Input layer is signaled to continue resonation and hence block any input until either a category is matched or a new one is created. Additionally, the category which failed the vigilance test is prohibited from competing with the current input until either another category passes the test, or a new category is created.

#### 4.3.8 Functional Layer

The Functional layer is a series of services performed on the final Category layer. These services are: Hypervolume Measure, Hypervolume Test, Hypervolume Adjust, Hypercube Learning, and Hypercube Creation.

**Hypervolume Measure.** The hypervolume  $hv$  is calculated by the product of the LTM weights as below:

$$hv = \prod_{i=1}^N (W_i - V_i) \quad (36)$$

**Hypervolume Test.** The overall hypervolume of each hypercube is maintained within bounds in order to keep the hypercubes from expanding to infinite volume. The limit is essentially a bound for learning in the network. The volume parameter is defined as follows:

$$\sum (W_i - V_i) \leq volume \quad \text{or} \quad n\Delta \leq volume \quad (37)$$

The value  $N$  is the number of input nodes and  $\Delta$  the hypervolume per node. The hypervolume limit testing and adjustment is necessary since each of the dimensions of a

hypercube are not constrained, such as in the case of fuzzy ART where the weights must be strictly decreasing. In [25], the hypervolume limits on the categories are limited to the unit hypercube as follows:

$$\sum_{i=1}^N \left( \max(w_{ji}, a_{hi}) - \min(v_{ji}, a_{hi}) \right) \leq \Theta, \quad (38)$$

$$0 \leq \Theta \leq 1$$

The problem with the hypervolume test above is that it cannot easily accommodate “noisy” hypercubes in the category layer. Since this is a problem with the basic fuzzy ART, the limit must be changed to allow for noisy data. In the case of the fuzzy hypercube ART network, the hypercube volume is constrained to be less than a maximum limit,  $hv_{\max}$ ,

$$\Theta \leq hv_{\max} \quad (39)$$

An additional parameter is defined, hypercube dimension,  $hd_{\max}$ , assuming equal size in each dimension:

$$hd_{\max} = \frac{hv_{\max}}{N} \quad (40)$$

**Hypervolume Adjust.** If the limit  $hv_{\max}$  is exceeded, the entire hypervolume is adjusted to maintain inequality (Eq.42b). The excessive volume  $\Delta hv$  is found from the current hypervolume,  $hv$ , by the following:

$$\Delta hv = \begin{cases} (hv - hv_{\max}) / N & hv > hv_{\max} \\ 0 & \text{Otherwise} \end{cases} \quad (41)$$

where  $N$  is the input dimensionality. The hypervolume of the current category  $[J]$  must be adjusted whenever  $\Delta hv > 0$  by

$$\begin{aligned} W_J^{new} &= \max\{(W_J^{old} - \Delta hv), 0\} \\ V_J^{new} &= \min\{(V_J^{old} + \Delta hv), 1\} \end{aligned} \quad (42)$$

This operation brings the hypervolume of each selected category within the value of  $hv_{\max}$ .

**Hypercube Learning.** The inclusion of input  $A_i$  into the winning category hypercube  $B_j$  is done through a learning algorithm which adjusts the hypercube of category  $[J]$ . In general, each value of  $A_i$ , selectively adjusts its respective limits in  $W_j$  and  $V_j$ .

Given an input vector  $A_i$  and a hypercube  $B_j$ , and a learning adjustment factor  $r$ , learning on a case by case basis is performed for each dimension of  $A$  over the entire chosen category  $B_j$  as follows:

Case 1: Initialization.

$$\begin{aligned} W_{ji}^{new} &= V_{ji}^{new} = A_i \\ \text{whenever } W_{ji}^{old} &\leq A_i \text{ and } V_{ji}^{old} \geq A_i \end{aligned} \quad (43a)$$

Case 2: Input is above  $W_j$

$$W_{ji}^{new} = W_{ji}^{old} (1 - r) + rA_i \quad (43b)$$

*whenever*  $A_i > W_{ji}^{old}$  *and*  $A_j \leq V_{ji}^{old} + hd_{\max}$

Case 3: Input is below  $V_j$

$$V_{ji}^{new} = V_{ji}^{old} (1 - r) + rA_i \quad (43c)$$

*whenever*  $A_i < V_{ji}^{old}$  *and*  $A_i \geq W_{ji}^{old} - hd_{\max}$

Case 4: Input is within  $B_j$ .

$$\text{Whenever } A_{ii} \geq V_{ji}^{old} \quad \text{and} \quad A_i \leq W_{ji}^{old}:$$

4a) Input is closer to  $W$

$$W_{ji}^{new} = W_{ji}^{old} (1 - r) - rA_i \quad (43d)$$

*whenever*  $W_{ji}^{old} - A_i > A_i - V_{ji}^{old}$

4b) Input is closer to  $V$

$$V_{ji}^{new} = V_{ji}^{old} (1 - r) + rA_i \quad (43e)$$

*whenever*  $A_i - V_{ji}^{old} > W_{ji}^{old} - A_i$

Case 5:

$$V_{ji}^{new} = A_i - W_{ji}^{old} - hd_{\max} \quad (43f)$$

*whenever*  $A_i > W_{ji}^{old}$  *and*  $A_i < V_{ji}^{old} + hd_{\max}$

Case 6:

$$W_{ji}^{new} = V_{ji}^{old} - A_i + hd_{\max} \quad (43g)$$

*whenever*  $A_i < V_{ji}^{old}$  *and*  $A_i > W_{ji}^{old} - hd_{\max}$

**Hypercube Creation.** The creation of a hypercube requires that the overall hypervolume limit is adjusted through the hypercube dimension,  $hd_{\max}$ , which depends on the number of categories in the network,  $N$ , from equation 30.

#### 4.3.9 Category Layer

The Category layer consists of a set of complex neurons with associated states and LTM weight values which describe them. The LTM weights are associated with the min-max feature hypercube representation of the associated  $J$ -categories defined by Simpson [25]. Each hypercube category  $C$  is a fuzzy cluster defined by:

$$C_j = \{B_j, N^j, T^j, S^j\}$$

$$B_j = \{V_j, W_j\}, \quad j = 1, 2, \dots, N_{\max} \quad (44)$$

$$V_j, W_j \in [0, 1]$$

where  $N^j$  is the count of adjustments,  $T^j$  is the confidence and  $S^j$  is the state of category  $[j]$ .  $B_j$  is the hypercube representation of category  $j$ ,  $V_j$  is the minimum point,  $W_j$  the maximum point, and  $N_{\max}$  the total number of categories.

## 4.4 Category Merge

A global merge is defined as the combination of cluster classes produced by the neural network which are very "close" to one another. This operation is performed outside of the neural network processing and does not affect any of the internal operation of the network. It does, however, utilize detail parameters generated by the network, and hence can be considered a higher order operation of the network which is bound to its operation. This process also occurs over time between NN cycles and can be considered a long-term-averaging process.

### 4.4.1 Merge Parameters

There are two measures which are used to indicate whether a global merge is to take place:

- a) Volume difference between hypercube categories
- b) Magnitude of rating R from equation (41) between two categories.

### 4.4.2 Merge Criteria

A function is defined which performs the category merge. First, the merge parameters are obtained over all possible different pairs of the current categories defined. Next, the merge criteria are applied and used to partition the current categories into a final set of categories which is compacted using the criteria. Note that the compacting occurred very rarely during testing.

The criteria are expressed in terms of acceptance/rejection regions in the volume difference/rating mapping.

$$\begin{aligned} 0.0 \leq \Delta vol(c1, c2) \leq 1.10 \text{ and } R(c1, c2) > 1.00 \quad OR \\ 1.1 < \Delta vol(c1, c2) \leq 1.50 \text{ and } R(c1, c2) > 1.00 \quad OR \\ 1.5 < \Delta vol(c1, c2) \leq 1.75 \text{ and } R(c1, c2) > 1.40 \end{aligned} \quad (45)$$

These were experimentally derived and were only used to evaluate the concept of global clustering criteria within the context of the hypercube structure.

## 4.5 Initialization

The initialization is performed on the network as follows.

I.1 Enable all categories, set count, and confidence is "none".

$$N^J = 0, \quad T^J = \text{none}, \quad S^J = \text{enabled} \quad (46a)$$

I.2 Set all categories

$$V_{ji}^{new} = 1, \quad W_{ji}^{new} = 0, \quad \text{for } i = 1, 2, \dots, N \quad J = 1, 2, \dots, N_{\max} \quad (46b)$$

## 5. TEST METHODOLOGY

### 5.1 Test Data

There were two data sets used for the formal testing of the system, the Switchboard [27] and the Greenflag [28].

	Spkr 1	Spkr 2	Spkr 3	Spkr 4	Spkr 5	Spkr 6	Spkr 7	Spkr 8	M/F
Set 1	02	15	38	46	62	81	28	33	6/2
Set 2	04	41	72	02	15	38	46	62	5/3
Set 3	05	23	27	42	56	59	17	32	4/4
Set 4	15	38	46	62	81	28	33	76	7/1
Set 5 (1)	04	41	72	02	15	38	46	62	
Set 5 (2)	32	35	65	90	39	82			10/6
Set 6 (1)	15	38	46	66	04	41	72	02	
Set 6 (2)	65	90	39	82					
Set 7	57	17	32	35	65	90	39	82	5/3
Set 8	72	02	15	38	46	62	81	28	5/3
Set 9	90	39	82	04	41	72	02	15	4/4

Table 1: Switchboard 95 Test Set to Actual Speaker Reference

	Spkr 1	Spkr 2	Spkr 3	Spkr 4	Spkr 5	Spkr 6	Spkr 7	Spkr 8
Set 1	ccz	ccv	cdi	cdk	ccd	cch	cdn	cdt
Set 2	cdw	cfp	cfj	cel	cev	cfs	cfu	cfx
Set 3	cfi	ccv	cdi	cdk	ccd	cin	cdn	cdt
Set 4	cga	cfp	cfj	cel	cev	cfs	cfu	cfx
Set 5	cgm	chs	ckd	chc	cdk	chg	cch	ccz
Set 6	cgp	chs	ccd	chc	chy	chg	cfu	cfx
Set 7	cgx	chs	ccd	chc	cdk	chg	cfu	ccz
Set 8	chc	cel	cdc	cgx	ccd	cin	cdn	cdt
Set 9	chj	chs	chn	cii	cin	chg	cif	cik
Set 10 (1)	ccz	ccv	cdi	cdk	ccd	cch	cdn	cdt
Set 10 (2)	chj	chs	chn	cii	cin	chg	cif	cik
Set 11	chy	cel	cji	cgx	ccd	cin	cdn	ceb
Set 12	cdv	cen	ckb	cge	cin	ckd	cif	cic
Set 13	ckc	chs	ckb	cii	cin	chg	cif	cik
Set 14 (1)	ccz	ccv	cdi	cdk	ccd	cch	cdn	cdt
Set 14 (2)	chj	chs	chn	cii				
Set 15 (1)	cga	cfp	cfj	cel	cev	cfs	cfu	cfx
Set 15 (2)	cgm	chs	chc	chg				

Table 2: Greenflag Test Set to Actual Speaker Reference

#### 5.1.1 Switchboard data set.

The Switchboard data were grouped into sets of 8, 12, and 16 speakers. The actual breakout of the speakers is shown in Table 1. The vertical entries are the 9 speaker sets consisting of the actual speakers given by the file numbers of individual speakers in the

data set. Additionally, the numbers of male/female speakers is given in the last column. For a detailed description of the Switchboard database see [27].

### 5.1.2 Greenflag data set.

The Greenflag test data were organized the same as the Switchboard, except that there are 15 sets. The set ID's are given by three letter combinations all beginning with a "c". See Table 2, Greenflag Test Set to Actual Speaker Reference, for the breakout. For a detailed description of the Greenflag database see [28].

## 5.2 Test Conditions

The subsystems of Feature Processing, Feature Preprocessing, and Neural Network Classifier were tested using the test data described in section 5.1. There were a number of fixed and varied parameters corresponding to specific subsystems as given below.

All the below parameters are specifically related to the hypercube network. The basic ART and fuzzy ART have different parameters and are so indicated below.

#### a) Feature Processing: fixed parameters

- Mel Cepstral Parameters = 13
- Reflection Coefficients = 12
- LPC Cepstral Coefficients = 7
- Correlation Order [ $O_{corr} = 13$ ]
- Number of LPC poles [ $p = 14$ ]
- Number of Mel bands [ $N_{bands} = 12$ ]
- Max/Min values of Features [see Table 3]

	$f0$	$f1$	$f2$	$f3$	$f4$	$f5$	$f6$	$f7$	$f8$	$f9$	$f10$	$f11$	$f12$
LPC cepstrum Maximum		-0.2	0	0	0	0	0	0					
Mel cepstrum Maximum	71.3	6.9	-1.5	9.5	-2.0	8.0	-1.0	6.6	-0.6	5.0	-0.3	2.7	0.1
Reflection Maximum		0.5	0.45	0.25	0.25	0.2	0.2	0.18	0.17	0.17	0.22	0.17	0.19
LPC cepstrum Minimum		-0.8	-0.4	-0.25	-0.2	-0.2	-0.1	-0.1					
Mel cepstrum Minimum	65.0	2.7	-5.0	7.0	-4.5	6.0	-3.0	5.4	-2.0	3.6	-1.5	2.0	-0.6
Reflection Minimum		-0.057	-0.11	-0.056	0.1	0.07	0	0	-0.01	-0.012	-0.05	0.02	-0.051

**Table 3: Maximum/Minimum Values of Features**

#### b) Signal Segmentation: variable parameters

- Total Number of Segments of Voice Speech Processed
- Average Time per Voiced Speech segment
- Minimum Time per Voiced Speech Segment

#### c) Signal Segmentation and Voiced/Unvoiced: fixed parameters

- Number of samples per segment [ $N_{sam} = 128$ ]
- AMDF fraction of samples per chunk [0.75]
- Minimum and maximum pitch [ $P_{min} = 1.9, P_{max} = 18.0$ ]
- Minimum and maximum zero crossing frequency [ $n_{min} 0.6, n_{max} = 5.0$  in Khz]

Minimum voiced energy threshold [ $E_{\min} = 1000$ ]

d) Feature Preprocessing

Rule base for:

IF (Proportional # matched segments is N1)

AND (Proportional # linked segments is N2)

THEN (Hypothesis Truth that segment S represents a valid speaker is V)

e) Neural Network: fixed parameters

e1) Fuzzy Hypercube/Fuzzy ART:

Maximum number of attributes in a pattern [NN\_MAXATTR = 50]

Maximum number of opinions per pass [NN\_MAXOPINIONS = 2]

Maximum number of class that may be formed [NN\_MAXCLASS = 50]

Lower limit, upper limit initialization value [LL\_Init = 1.0, UL\_Init = 0.0]

e2) Basic ART:

Maximum number of attributes in a pattern [NN\_MAXATTR = 200]

Maximum number of opinions per pass [NN\_MAXOPINIONS = 2]

Feedback from top layer [NN\_TOPDOWN\_FEEDBACK = 0.8]

Maximum number of pattern identifier [NN\_MAXID = 20]

Degree of functional expansion [NN\_FUNC\_EXPAND = 10]

Lower limit, upper limit initialization value [LL\_Init = 0.0, UL\_Init = 1.0]

f) Neural Network: variable parameters

Vigilance

Maximum Hypervolume

g) Overall System variable data & parameters

Test Data Sets

Number of Speakers

General ART vs Fuzzy ART vs Fuzzy Hypercube ART

Number of correct & incorrect classifications per Test Set

## 5.3 Test Results

The test data in 5.1 were applied according to the test conditions of section 5.2, and the results are reported in this section. There were several parametric tests; measurements were made on each test run in the following sections.

### 5.3.1 Feature Processing

Features were analyzed for two basic characteristics, separability, and maximum/minimum values. The separability were observed using the XGOBI visualization tool. It allows a multidimensional viewing of the features and their clustering ability. The max/min values were determined from a basic test set not included in the test results.

### 5.3.2 Neural Network

The following are parameters which were varied in the neural network during the testing:

Total Number of Actual Speakers Correctly Identified ( $i_1$ ),  $\geq 1$  class per node

Total Number of Invalid HCNs generated ( $i_2$ )

Total Number of Invalid HCNs generated ( $i_3$ )

Total Number of Invalid HCN's generated ( $i_4$ )

The value of  $i_1$  is a count of the correct number of HCN's generated by the NN which corresponds to real speakers. This gives a number of the correct number of categories generated, independent of the number of data sets presented to the network. The value of  $i_2$  is a count of the number of HCN's generated by the NN which are in addition to the set  $i_1$ .

$$\{HCN\} = \sum_{all\ HCN's} \{i_1 + i_2\} \quad (47)$$

$$\{i_1\} \cap \{i_2\} = \emptyset$$

where HCN is a set of hypercube category nodes generated during a complete test run,  $i_1$  is a count of correct HCN's and  $i_2$  is the incorrect HCN's count. Table 4 and Table 6 both display the results of  $I_1$  as a function of the vigilance parameter and the maximum hypervolume within a small range of values.

The values of  $i_3$  and  $i_4$  are spurious nodes generated and count of data sets in the spurious nodes. These values do not affect the values of correct/incorrect classification since they generally consist of nodes with only one or two entries, which is the definition of a spurious node. Table 5 and Table 7 display the spurious category creation in the network as a function of vigilance parameter and maximum hypervolume, again within the same small range of values.

Summarized test results for the fuzzy hypercube neural network performance are shown in Table 5.

<i>Test Data Set for 8 Speakers</i>	<i>Total Number of Speakers in Test</i>	<i>Total Voiced Speaking Time (hrs)</i>	<i>Overall Correct Classification (%)</i>
Switchboard May 95	26	2.69	69.7
Greenflag	41	2.96	70.3

**TABLE 4: Test Results for 8-Speaker Group**

### 5.3.3 Overall System

Parameters which are a measure of the overall system are given in this section.

Number of correct and incorrect classifications per Test Set

Total time per Test Set

The average overall percent correct classification is defined by:

$$P_c = \sum_{\text{over } M \text{ test sets}} \frac{\left( \frac{\sum_{\text{test set } m} \left( \frac{TD2_m}{TD2_m + TD3_m} \right)}{\sum m} \right)}{M} \quad (48)$$

The numerator of the summation of Eq. 37 is the mean of each individual test performed, while the exterior summation averages all the average classification fractions.

A series of tests which used the value of Eq. 37 were performed. First, two basic tests were run to evaluate the effects of the vigilance and hypervolume limit on  $P_c$ . These are shown in Tables 3 and 4. The absolute values of minimum and maximum obtained during the entire test period are shown in relation to the mean value which is plotted against the vigilance and hypervolume limit values.

The generation of the correct (C) and incorrect (I) classifications are related to the neural network values I1-I4, but were visually chosen from these sets as the values which provided the greatest correct classifications per HCN. This would require a simple program, which has the maximum number of entries as correct nodes, to choose the distinct nodes. Also, the totals generated under the neural network required additional data analysis and speaker truth.

The summarized results for the overall Switchboard and Greenflag data sets taken for 8 and 12 speakers are given in Tables 6 and 7.

<i>Test Data Set for 12 Speakers</i>	<i>Total Number of Speakers in Test</i>	<i>Total Voiced Speaking Time (mins)</i>	<i>Overall Correct Classification (%)</i>
Switchboard May 95	12	16.71	67.25
Greenflag	23	4.77	68.75

**TABLE 5: Test Results for 12-Speaker Group**

<i>Test Data Set for 8 Speakers</i>	<i>Average Number of Correct Categories Generated (8 max)</i>	<i>Average Number of False Categories Generated (8 max)</i>	<i>Average Number of False Categories Deleted per Data Set</i>
Switchboard May 95	6.29	0.29	1.86
Greenflag	6.57	0.23	5.77

**TABLE 6: Fuzzy Hypercube Neural Network Test Results**

The overall system test results are shown in Table 7. This includes all speaker groups.

<i>Test Data Set for all Speaker Groups</i>	<i>Total Number of Speakers</i>	<i>Total Voiced Speaking Time (hrs)</i>	<i>Overall Correct Classification (%)</i>	<i>Standard Deviation (avg)</i>	<i>Maximum- Minimum (avg)</i>
Switchboard May 95	26	3	66.9	5.0	14.5
Greenflag	41	3	66.6	6.6	13.4

**TABLE 7: Overall System Test Results**

## 6. DISCUSSION

### 6.1 Overall

The Overall testing results are shown in Tables 1, 2, 5 and 6. The results are synopsized in Table 7, giving the standard deviation averaged over all groups for each group, as well as the maximum to minimum value spread averaged over all the groups.

From these data, it can be seen that Greenflag had a smaller minimum to maximum spread, and, with the exception of group #7, all appear well behaved. In the switchboard case, the spread is much more in all groups with number 13 the greatest. However, the switchboard data were still more well behaved and better clustered as is shown by their better standard deviation value shown in Table 7.

The performance of the test groups is nearly identical at 67%, but this is for an 8 speaker group maximum.

### 6.2 Recommendations for Future Research and Improvements

The recommendations for improving the current system with changes and additional areas of research are presented for the features, classifier, and overall system. The following are areas that can be investigated for improvement to the speaker recognition process:

- a) Inclusion of new features. The inclusion of new features is a constant improvement which can be made to the Speaker Recognition System feature processing. Some of the features which may be of use are:
  1. Delta Cepstrum
  2. Cepstrum with mean removal
  3. Log Energy
  4. Average Pitch
  5. RASTA/PLP
- b) Expansion of input through complementation
- c) Inclusion of listener models
- e) Inclusion of specific verbal cue modeling for specific languages.

## 7. BIBLIOGRAPHY

### 7.1 Technical References

- [1] Atal, Bishnu S., "Automatic Recognition of Speakers from Their Voices," Proc. of the IEEE, Vol. 64, No. 4, pp. 460-475 (April 1976).
- [2] Buckley, J.J., W. Siler, D. Tucker, "A Fuzzy Expert System," Fuzzy Sets and Systems, Vol. 20, pp. 1-16 (1986).
- [3] Carpenter, Gail A. and Stephen Grossberg, "ART2: Self-organization of Stable Category Recognition Codes for Analog Input Patterns," Applied Optics, Vol. 26, No. 23, pp. 4919-4930 (1987).
- [4] Carpenter, Gail A. and Stephen Grossberg, "ART3: Hierarchical Search Using Chemical Transmitters in Self-Organizing Pattern Recognition Architectures," Neural Networks, Vol. 3, pp. 129-152 (1990).
- [5] Carpenter, Gail A. et al., "ART and ARTMAP Neural Networks for Applications: Self-Organizing Learning, Recognition, and Prediction," Tech Report CAS/CNS-96-009 (1996).
- [6] Carpenter, Gail A and Stephen Grossberg, "Learning, Categorization, Rule Formation, and Prediction by Fuzzy Neural Networks," Tech Report CAS/CNS-94-028 (1994)
- [7] Carpenter, Gail A., S. Grossberg and D. Rosen, "Fuzzy ART: An Adaptive Resonance Algorithm for Rapid, Stable Classification of Analog Patterns," Tech Report CAS/CNS-TR-91-006 (1991).
- [8] Carpenter, Gail A., M. Gajda, "Fuzzy ART Choice Functions," CAS/CNS-TR-93-060 (1993).
- [9] Colombi, J., "Cepstral and Auditory Model Features For Speaker Recognition," AFIT, 92D-11, AD-A259076 (1992).
- [10] Eck, J. Thomas, "An Enhanced ART2 Architecture and an Application to Automatic Speaker Recognition," Department of Computer Information Science, New Jersey Institute of Technology (1991).
- [11] Hermansky, H. et al., "RASTA-PLP Speech Analysis Technique," Proc. ICASSP (1993).
- [12] Karakowski, J., "An Automatic Text-Independent Speaker Recognition System," 26th Asilomar Conference on Signals, Systems, and Computers (1992).

- [13] Karakowski, J. et al., "Communication Net Sorting: An Automatic Text Independent Speaker Recognition System," US Army CECOM Report CECOM/EW-TR- 92-2 (1992).
- [14] Kasuba, T., "Simplified Fuzzy ARTMAP," AI Expert (Nov,1993).
- [15] Krishnapuram R., J.M., Keller, " A Possibilistic Approach to Clustering," IEEE Trans. Fuzzy Systems, Vol.1 No. 2 (1993).
- [16] Markel, J.D. and A.H. Gray, "Linear Prediction of Speech," Springer-Verlag, New York (1976).
- [17] Markel, J.D., S. B. Davis, "Text-Independent Speaker Recognition from a Large Linguistically Unconstrained Time-Spaced Database," IEEE Trans. on Acoustics, Speech, and Signal Processing, Vol. ASSP-27, No. 1 (1979).
- [18] Peacocke, R.D., D. H. Graf, "An Introduction to Speech and Speaker Recognition," Computer, pp. 26-33 (Aug 1990).
- [19] Pellisier, S.V., "Open-Set, Text-Independent Speaker Recognition," Thesis, AFIT/GE/ENG/95D, Air Force Institute of Technology, WPAFB, Ohio (1995).
- [20] Reynolds, D., "Experimental Evaluation of Features for Robust Speaker Identification," IEEE Trans. SAP, Vol. 2, No. 4 (Oct 1994).
- [21] Ross, Myron J. et al., "Average Magnitude Difference Function Pitch Extractor," IEEE Trans. on Acoustics, Speech, and Signal Processing, Vol. ASSP-22, No. 5, pp. 353-362 (Oct 1974).
- [22] Schafer, Ronald W. and Lawrence R. Rabiner, "Digital Representations of Speech Signals," Proc. of the IEEE, Vol. 63, No. 4, pp. 662-677 (Apr 1975).
- [23] Shih, Frank Y. and Jenlong Moh, "Improved Adaptive Resonance Theory," Proc. SPIE Conf. on Intelligent Robots and Computer Vision IX: Neural, Biological, and 3-D Methods (Nov. 1990).
- [24] Shih, Frank Y., Jenlong Moh, and Henry Bourne, "A Neural Architecture Applied to the Enhancement of Noisy Binary Images without Prior Knowledge," Proc. IEEE Intl. Conf. on Tools for Artificial Intelligence (Nov. 1990).
- [25] Simpson, P.K., "Fuzzy Min-Max Neural Networks-PART2: Clustering," IEEE Trans. Fuzzy Systems, Vol.1, No. 1 (Feb 1993).
- [26] Soong, F., Rosenberg, A., "On the Use of Instantaneous and Transitional Spectral Information in Speaker Recognition," IEEE Trans SAP, Vol 36, No 6 (June 1988)

## **7.2 Data Base References**

- [27] "Switchboard Speaker Evaluation-Training Segments & Conversation Sides and Test Target & Background Segments," National Institute of Standards and Technology, Gaithersburg, MD (May 1995).
- [28] Rome Air Development Center, Greenflag Data Base.

## Distribution List

Defense Technical Information Center  
ATTN: DTIC-OCC  
8725 John J. Kingman Rd., STE 0944  
Fort Belvoir, VA 22060-6218  
(\*Note: 2 DTIC copies will be sent  
from STINFO Office, Ft Monmouth, NJ)

Commander, U.S. Army CECOM  
R&D Technical Library  
Fort Monmouth, NJ 07703-5703  
(1) AMSEL-IM-BM-I-L-R (Tech Lib)  
(2) AMSEL-IM-BM-I-L-R (STINFO)

Commander, U.S. Army CECOM  
Research, Development and Eng. Center  
ATTN: AMSEL-RD  
Fort Monmouth, NJ 07703-5000

Commander, U.S. Army CECOM  
Director, C2SID  
ATTN: AMSEL-RD-C2-ED  
Fort Monmouth, NJ 07703

Commander, AFRL/CC  
1864 4th Street, Suit 1  
WPAFB, OH 45433

Director, Rome Research Site  
26 Electronic Parkway, Building 106  
Rome, NY 13442-4514

Commander, Naval Research Laboratory  
Code 1000  
Washington DC 20375-5320

Commander, AFIT/CC  
2950 P Street  
WPAFB, OH 45433-6583